

**Šakotieji
algoritmai.
Sąlygos tikrinimo
sakiny
(*if* operatorius)**

- Susipažinsite su šakotaisiais algoritmais ir sąlygos tikrinimo operatoriumi *if*.
- Išmoksite sąlygos tikrinimo operatorių *if* panaudoti realizuojant šakotuosius skaičiavimus.

TEORINĖ DALIS

Dažnai tenka spręsti uždavinius, kai reikia pasirinkti vieną veiksmą iš dviejų galimų. Sudėtingesniuose uždaviniuose reikia pasirinkti vieną veiksmą iš kelių ar net keliolikos galimų. Tam naudojame sąlygos tikrinimo operatorių *if*. Jo bendra forma:

if (Sąlyga)

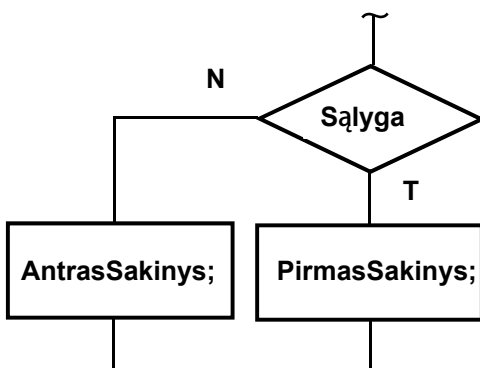
PirmasSakinys;

else

AntrasSakinys;

Jei sąlyga tenkinama, vykdomas **PirmasSakinys**. Priešingu atveju – **AntrasSakinys**.

Schematiškai tai galima pavaizduoti:



Sąlyga – tai reiškinys, kuriame naudojamos aritmetinės ir palyginimo operacijos. **ARITMETINĖS Palyginimo OPERACIJOS:**

$>$, $>=$, $<$, $<=$, $==(lygu)$, $!=(nelygu)$.

Reiškinys įgauna vieną iš dviejų galimų reikšmių: *true*(tiesa – **T**) arba *false*(netiesa – **N**).

Pavyzdžiui, $x + y > z$, $a == 1$, $b != c$, $j > 1$, $i <= 20$.

Sąlygose gali būti naudojamos ir **LOGINĖS OPERACIJOS &&, ||, !**. Su jomis jūs susipažinsite vėliau.

Kiekvienoje šakoje gali būti ir daugiau nei vienas sakiny. Tada sakiniai skliaudžiami operatoriniais skliaustais **{ }**.

if (Sąlyga)

{

TaipSakinys1;

TaipSakinys2;

...

TaipSakinysn;

}

else

{

NeSakinys1;

NeSakinys2;

...

NeSakinysm;

}

Tikrinant programos teisingumą, reikia patikrinti visas šakas. Realiuose uždaviniuose kartais taip padaryti labai sudėtinga, nes yra didelis šakų kiekis.

1 PAVYZDYS. Ant stalo buvo padėtos dvi saldainių dėžutės. Rita ir Siga paėmė dėžutėse esančius saldainius. Ar merginos turi vienodą saldainių kiekį?

Sprendimas

```
int kiekis1, kiekis2;
Console.WriteLine("Įveskite Ritos saldainių kiekį");
kiekis1 = int.Parse(Console.ReadLine());
Console.WriteLine("Įveskite Sigos saldainių kiekį");
kiekis2 = int.Parse(Console.ReadLine());
if (kiekis1 == kiekis2)
    Console.WriteLine("Rita ir Siga turi vienodą saldainių kiekį");
else
    Console.WriteLine("Rita ir Siga turi nevienodą saldainių kiekį");
```

Įvykdome programą panaudodami sąlygos tikrinimo operatorių *if*. Gauname atsakymus.

```
Įveskite Ritos saldainių kiekį
15
Įveskite Sigos saldainių kiekį
13
Rita ir Siga turi nevienodą saldainių kiekį
Press any key to continue . . .
```

Tą pačią užduotį išspręskite dar kartą. Sąlygos tikrinimo operatoriuje *if* panaudokime lyginimo operatorių *!=(nelygu)*. Sukeiskime **Taip** ir **Ne** šakas vietomis.

```
int kiekis1, kiekis2;
Console.WriteLine("Įveskite Ritos saldainių kiekį");
kiekis1 = int.Parse(Console.ReadLine());
Console.WriteLine("Įveskite Sigos saldainių kiekį");
kiekis2 = int.Parse(Console.ReadLine());
if (kiekis1 != kiekis2)
    Console.WriteLine("Rita ir Siga turi nevienodą saldainių kiekį");
else
    Console.WriteLine("Rita ir Siga turi vienodą saldainių kiekį");
```

Įvykdome programą. Gauname tą patį atsakymą.

```
Įveskite Ritos saldainių kiekį
15
Įveskite Sigos saldainių kiekį
13
Rita ir Siga turi nevienodą saldainių kiekį
Press any key to continue . . .
```

2 PAVYZDYS. Ant stalo buvo padėtos dvi saldainių dėžutės. Žinoma, kad dėžutėse yra skirtingas saldainių kiekis. Rita ir Siga paėmė dėžutėse esančius saldainius. Kuri mergina turi daugiau saldainių?

Sprendimas

```
int kiekis1, kiekis2;
Console.WriteLine("Įveskite Ritos saldainių kiekį.");
kiekis1 = int.Parse(Console.ReadLine());
Console.WriteLine("Įveskite Sigos saldainių kiekį.");
kiekis2 = int.Parse(Console.ReadLine());
if (kiekis1 > kiekis2)
    Console.WriteLine("Rita turi daugiau saldainių nei Siga.");
else
    Console.WriteLine("Siga turi daugiau saldainių nei Rita.");
```

Įvykdome programą panaudodami sąlygos tikrinimo operatorių *if*. Gauname atsakymus.

```
Įveskite Ritos saldainių kiekį.
15
Įveskite Sigos saldainių kiekį.
13
Rita turi daugiau saldainių nei Siga.
Press any key to continue . . .
```

3 PAVYZDYS. 2 pavyzdžio uždutį išspręsimė dar kartą. Tik kitaip užrašysime tikrinimo sąlygą.

Sprendimas. Užrašant sąlygos tikrinimo operatorių, galima keisti palyginimo ženklą priešingu. Taip padarius, sakiniai, esantys **Taip** ir **Ne** šakose, susikeičia vietomis. Spręsdami **2 pavyzdžio** sąlygos tikrinime panaudojome ženklą $>$. Jam priešingas ženklas yra $<=$. Sąlygoje duota, kad kiekiai yra skirtingi. Todėl ženklą $>$ galima keisti ir ženklu $<$.

```
int kiekis1, kiekis2;
Console.WriteLine("Įveskite Ritos saldainių kiekį.");
kiekis1 = int.Parse(Console.ReadLine());
Console.WriteLine("Įveskite Sigos saldainių kiekį.");
kiekis2 = int.Parse(Console.ReadLine());
if (kiekis1 < kiekis2)
    Console.WriteLine("Siga turi daugiau saldainių nei Rita.");
else
    Console.WriteLine("Rita turi daugiau saldainių nei Siga.");
```

Įvykdome pakeistą programą. Gauname atsakymus.

```
Įveskite Ritos saldainių kiekį.
15
Įveskite Sigos saldainių kiekį.
13
Rita turi daugiau saldainių nei Siga.
Press any key to continue . . .
```

Kaip matote, atsakymai nepakito.

4 PAVYZDYS. Duoti du skirtingi svoriai. Didesnį sumažinti 5 svorio vienetais, o mažesnį tiek pat padidinti.

Sprendimas

```
int svoris1, svoris2;
svoris1 = 10;
svoris2 = 20;
Console.WriteLine("Pradiniai svoriai:");
Console.WriteLine("svoris1 = {0} svoris2 = {1}", svoris1, svoris2);
if (svoris1 > svoris2)
{
    svoris1 -= 5;
    svoris2 += 5;
}
else
{
    svoris1 += 5;
    svoris2 -= 5;
}
Console.WriteLine("Nauji svoriai:");
Console.WriteLine("svoris1 = {0} svoris2 = {1}", svoris1, svoris2);
```

Kaip matote, kiekvienoje šakoje yra po 2 veiksmus: vienas svoris mažinamas, kitas – didinamas. Todėl būtina naudoti operatorinius skliaustus { }. Įvykdome programą. Gauname atsakymus.

```
Pradiniai svoriai:
svoris1 = 10 svoris2 = 20
Nauji svoriai:
svoris1 = 15 svoris2 = 15
Press any key to continue . . .
```

5 PAVYZDYS. Duoti du svoriai. Jei pirmas svoris yra didesnis, atspausdinkite pranešimą keliais vienetais šį svorį reikia sumažinti, kad abu svoriai taptų vienodais.

Sprendimas

```
int svoris1, svoris2;
svoris1 = 20;
svoris2 = 12;
Console.WriteLine("Pradiniai svoriai:");
Console.WriteLine("svoris1 = {0} svoris2 = {1}", svoris1, svoris2);
if (svoris1 > svoris2)
    Console.WriteLine("svoris1 reikia sumažinti {0} vienetais.",
        svoris1 - svoris2);
```

Įvykdome programą. Gauname atsakymus.

```
Pradiniai svoriai:
svoris1 = 20 svoris2 = 12
svoris1 reikia sumažinti 8 vienetais.
Press any key to continue . . .
```

Iš sąlygos matyti, kad veiksmas atliekamas tik **Taip** šakoje, o **Ne** šaka lieka tuščia. Jos patartina iš viso nerašyti. Jei parašytume `else`;, atsirastų įspėjimas, kad yra galimai klaidingas operatorius. Tai tik įspėjimas, tai dėmesio nebūtina kreipti.

1 pastaba. Jei sprendimo metu gaunasi atvejis, kai **Taip** šaka yra tuščia, o veiksmai atliekami tik **Ne** šakoje, patartina ženklą keisti į priešingą ir veiksmus atlikti **Taip** šakoje.

6 PAVYZDYS. Duoti du svoriai. Jei pirmas svoris nėra didesnis, atspausdinkite pranešimą keliais vienetais šį svorį reikia padidinti, kad abu svoriai taptų vienodais.

Sprendimas

```
int svoris1, svoris2;
svoris1 = 2;
svoris2 = 12;
Console.WriteLine("Pradiniai svoriai:");
Console.WriteLine("svoris1 = {0} svoris2 = {1}", svoris1, svoris2);
if (svoris1 > svoris2)
    ; // Tuščia šaka
else
    Console.WriteLine("svoris1 reikia padidinti {0} vienetais.",
        svoris2 - svoris1);
```

Įvykdome programą. Gauname atsakymus. Bet Visual Studio kompiliatorius išduoda įspėjimą dėl tuščios šakos.

```
Pradiniai svoriai:
svoris1 = 2 svoris2 = 12
svoris1 reikia padidinti 10 vienetais.
Press any key to continue . . .
```

Perrašome *if* operatoriaus sąlygą, pakeisdami sąlygos ženklą priešingu. Pakeista programa duoda tokį patį atsakymą, bet jau nėra įspėjimo. Ar ženklą keisti, ar ne – pasirenka autorius. Žemiau pateikta programa su pakeista *if* operatoriaus sąlyga.

```
int svoris1, svoris2;
svoris1 = 2;
svoris2 = 12;
Console.WriteLine("Pradiniai svoriai:");
Console.WriteLine("svoris1 = {0} svoris2 = {1}", svoris1, svoris2);
if (svoris1 <= svoris2)
    Console.WriteLine("svoris1 reikia padidinti {0} vienetais.",
        svoris2 - svoris1);
```

7 PAVYZDYS. Duoti trys svoriai: svoris1, svoris2, svoris3. Žinoma, kad jie visi yra skirtingi. Atspausdinkite didžiausio svorio vardą ir dydį.

Sprendimas

```
int svoris1, svoris2, svoris3;
svoris1 = 62;
svoris2 = 32;
svoris3 = 22;
Console.WriteLine("Pradiniai svoriai:");
Console.WriteLine("svoris1 = {0} svoris2 = {1} svoris3 = {2}",
    svoris1, svoris2, svoris3);
if (svoris1 > svoris2)
    if (svoris1 > svoris3)
        Console.WriteLine("Sunkiausias yra svoris1 = {0}.", svoris1);
    else
        Console.WriteLine("Sunkiausias yra svoris3 = {0}.", svoris3);
else
    if (svoris2 > svoris3)
        Console.WriteLine("Sunkiausias yra svoris2 = {0}.", svoris2);
    else
        Console.WriteLine("Sunkiausias yra svoris3 = {0}.", svoris3);
```

Šiame pavyzdyje turime 3 sąlygos tikrinimo operatorius *if*. Pirmo tikrinimo **Taip** ir **Ne** šakose turime po papildomą sąlygos tikrinimo operatorių. Jie reikalingi, kad atliktume visas galimas trijų dydžių porų palyginimus. Tik atlikę visų galimų porų palyginimus, turime surastą didžiausią svorį.

Tikrinant programos teisingumą, pirmuose 6 uždaviniuose pakanka 2 skirtingų testinių variantų, kad įrodytume, kad programa veikia teisingai. Yra tik po 2 skirtingas šakas. Paskutiniame uždavinyje jau reikia 4 skirtingų duomenų variantų teisingumui įrodyti. Tinkami variantai būtų:

svoris1 = 62; svoris2 = 32; svoris3 = 22.

svoris1 = 62; svoris2 = 32; svoris3 = 82.

svoris1 = 62; svoris2 = 72; svoris3 = 22.

svoris1 = 62; svoris2 = 72; svoris3 = 92.

Įvykdome programą su pirmu testiniu variantu. Gauname atsakymus.

```
Pradiniai svoriai:  
svoris1 = 62 svoris2 = 32 svoris3 = 22  
Šunkiausias yra svoris1 = 62.  
Press any key to continue . . .
```

8 PAVYZDYS. Duoti metai *m*. Atspausdinkite, ar jie yra keliamieji.

Sprendimas. Keliamieji metai yra tie, kurie be liekanos dalijasi iš 4, o paskutiniai amžiaus metai (dalijasi iš 100 be liekanos) yra keliamieji tik tada, jei jie be liekanos dalijasi iš 400. Todėl keliamieji metai yra 2000, 2012, 1912. Tačiau 1900, 1800, 1500 metai nėra keliamieji.

```
int m = 2000;  
if (m % 4 == 0)  
    if (m % 100 == 0)  
        if (m % 400 == 0)  
            Console.WriteLine("Metai m = {0} yra keliamieji.", m);  
        else  
            Console.WriteLine("Metai m = {0} nėra keliamieji.", m);  
    else  
        Console.WriteLine("Metai m = {0} yra keliamieji.", m);  
else  
    Console.WriteLine("Metai m = {0} nėra keliamieji.", m);
```

Įvykdome programą su vienu testiniu variantu. Gauname atsakymus.

```
Metai m = 2000 yra keliamieji.  
Press any key to continue . . .
```

Suskaičiuokite, kelių testinių variantų reikia pilnai patikrinti visas išsiskojimo šakas.

2 pastaba. Jei turime *n* skirtingų sąlygų, jas patikrinti pakanka $n - 1$ sąlygos tikrinimo operatoriaus.

9 PAVYZDYS. Apskaičiuokite kintamąjį y pagal formules:

$$y = \begin{cases} x + 1, & x < 1 \\ 2 * x - 5, & 1 \leq x \leq 6 \\ x^2, & x > 6 \end{cases}$$

Sprendimas. Iš sąlygos matyti, kad yra 3 skirtingi intervalai apskaičiuoti kintamajam y. Pasinaudojant **2 pastaba**, žinome, kad reikės dviejų tikrinimo operatorių *if*. Patikrinus sąlygą $x < 1$, **Taip** šakoje rašome pirmo intervalo formulę. Į **Ne** šaką automatiškai patenka visos x reikšmės, kurios netenkina sąlygos $x < 1$, t.y. $x \geq 1$.

Todėl tikrinant antrą intervalą $1 \leq x \leq 6$, sąlygos $x \geq 1$ tikrinti jau nebūtina. Pakanka patikrinti sąlygą $x \leq 6$. Išnagrinėjus pirmą ir antrą intervalus, lieka tik trečias. Sąlygos $x > 6$ tikrinti nebūtina, iš karto rašome formulę.

```
int x = 4;
int y;
if (x < 1)
    y = x + 1;
else if (x <= 6)
    y = 2 * x - 5;
else
    y = x * x;
Console.WriteLine(" x = {0} y = {1}", x, y);
```

Įvykdome programą su vienu testiniu variantu, kai $x = 4$. Gauname atsakymus.

```
x = 4 y = 3
Press any key to continue . . .
```

10 PAVYZDYS. Apskaičiuokite kintamąjį y pagal formules:

$$y = \begin{cases} x + 10, & \text{kai } x < 5 \text{ ir } y > 0; \\ x - 8, & \text{kai } x \geq 5. \end{cases}$$

Sprendimas. Sąlygoje duoti du intervalai kintamajam y skaičiuoti. Bet atidžiai pažiūrėjus į sąlygą matyti, kad egzistuoja dar vienas intervalas, kuriame y nereikia keisti. Tai intervalas $x < 5$ ir $y \leq 0$. Todėl patikrinus sąlygą $x < 5$, **Taip** šakoje naudosime operatorinius skliaustus **{ }** ir juose tikrinsime sąlygą $y > 0$ ir apskaičiuosime pirmą formulę. Skliaustuose esančiam *if* operatoriui nebūtina rašyti tuščios **Ne** šakos

```
int x = 4;
int y = 10;
if (x < 5)
{
    if (y > 0)
        y = x + 10;
}
else
    y = x - 8;
Console.WriteLine(" x = {0} y = {1}", x, y);
```

Įvykdome programą su vienu testiniu variantu, kai $x = 4$ ir $y = 10$. Gauname atsakymus.

```
x = 4 y = 14
Press any key to continue . . .
```

Šį uždavinį galima išspręsti ir kitaip, nenaudojant operatorinių skliaustų { }. Tada vidinio sąlygos tikrinimo operatoriaus `if` tuščią `else`; praleisti negalima.

3 pastaba. Skaičiavimuose nenaudojant operatorinių skliaustų { }, vidinių `if` operatorių tuščias `else`; praleisti galima tik tada, kai už jų yra praleistos **visos** išorinių `if else` šakos.

Išspendžiame uždavinį kitu būdu.

```
int x = 4;
int y = 10;
if (x < 5)
    if (y > 0)
        y = x + 10;
    else ; // praleisti negalima dėl sekančio else.
else
    y = x - 8;
Console.WriteLine(" x = {0} y = {1}", x, y);
```

Įvykdome programą su tuo pačiu testiniu variantu, kai $x = 4$ ir $y = 10$. Abiem atvejais atsakymai sutampa.

Pasirenkame kitą testinį variantą, kai $x = 4$ ir $y = -10$. Abiem atvejais atsakymai sutampa.

```
x = 4 y = -10
Press any key to continue . . .
```

Turint mažiau patirties tokio tipo uždaviniuose patartina naudoti operatorinius skliaustus { }.

11 PAVYZDYS. Apskaičiuokite kintamąjį y pagal formules:

$$y = \begin{cases} x + 2, & \text{kai } x \geq 0 \text{ ir } x \leq 10; \\ x - 1, & \text{kai } x < 0 \text{ arba } x > 10. \end{cases}$$

Sprendimas. Šį pavyzdį galima spręsti tokiu pat būdu, kaip ir **9 pavyzdys** ar **10 pavyzdys**. Bet `if` operatoriuje tikrinimo sąlyga gali būti ir sudėtinė, loginė sąlyga. Tokias sąlygas užrašyti jūs išmoksite vėliau.

Šiame pavyzdyje išnagrinėsime kaip užrašyti nesudėtingą sudėtinę sąlygą, t.y. sąlygą, kurią sudaro kelios paprastos sąlygos sujungtos loginiu veiksmu *ir* (žymima `&&`). Tokia išraiška reikšmę **Taip** įgyja tik tada, kai visos į ją įeinančios paprastos sąlygos yra **Taip**. Visais kitais atvejais reiškinio atsakymas yra **Ne**.

```
int x = 5;
int y;
if (x >= 0 && x <= 10)
    y = x + 2;
else
    y = x - 1;
Console.WriteLine(" x = {0} y = {1}", x, y);
```

Įvykdome programą su vienu testiniu variantu, kai $x = 5$. Kaip matyti loginė sąlyga $x \geq 0 \ \&\& \ x \leq 10$ įgyja reikšmę **Taip**. Gauname atsakymą.

```
x = 5 y = 7
Press any key to continue . . .
```